FORK-TOLERANT CONSENSUS PROTOCOL

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] This application is a divisional of U.S. patent application Ser. No. 16/378,456, filed Apr. 8, 2019, which claims the benefit of U.S. Provisional Application No. 62/655,175, filed Apr. 9, 2018, both of which are incorporated by reference.

BACKGROUND

1. Technical Field

[0002] The subject matter described generally relates to distributed ledgers, and in particular to a fork-tolerant consensus protocol for building a blockchain.

2. Background Information

[0003] Distributed ledgers were developed as a means for parties to engage in transactions, e.g., financial transactions, without the need for a single, trusted intermediary. In such systems, each transaction is recorded independently by several nodes (e.g., on a blockchain). No one entity controls all of the nodes so it is exceedingly difficult for a malicious actor to alter the transaction once it has been recorded by the nodes. Accordingly, the transactions can be conducted without the parties needing to trust each other, or any individual node provider.

[0004] More precisely, the participating nodes (who may not trust each other) can reach consensus and cooperate without a third party or a central authority. However, the decentralization used to replace individual trust typically involves the participation of a large number of diverse participants, impacting the throughput of the blockchain. Thus, practical deployments of blockchain networks have typically been faced with a choice between throughput and trust.

[0005] Another problem that arises with blockchains is relates to chain forks. In the blockchain, a miner or proposer releases the new block into the blockchain, but it takes some time before the producers converge on the longest chain. This results in chain forks and it is possible that a block released by a producer does not make it to the main chain at all, if that block is not harvested off the longest chain. Such orphaned blocks are either discarded or included in the chain to improve the security of the main chain. Either way, the transactions included in the orphaned blocks are ignored, thus wasting the block producer's time and effort in validating and including the transactions into those blocks. This wasteful computation happens in every round, especially when a large number of producers release new blocks concurrently. The very catalyst for decentralization contributes to this waste. While a large number of producers can create new blocks, only one of them will succeed in getting its block added to the main chain, thus severely limiting the blockchain throughput.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Figure (FIG. **1** is a block diagram illustrating a networked computing environment suitable for implementing a fork-tolerant consensus protocol, according to one example embodiment.

[0007] FIG. **2** illustrates the consensus network of FIG. **1**, according to one example embodiment.

[0008] FIG. **3**A illustrates the structure of a transaction message, according to one example embodiment.

[0009] FIG. **3**B illustrates a batch of transaction messages, according to one example embodiment.

[0010] FIG. **4**A illustrates a blockchain constructed using the fork-tolerant consensus protocol, according to one example embodiment.

[0011] FIG. **4**B illustrates a chain of transactions for an account, according to one example embodiment.

[0012] FIG. **5** is a flowchart illustrating a method for building a block using the fork-tolerant consensus protocol, according to one example embodiment.

[0013] FIG. **6** is a flowchart illustrating a method for a Validator to batch transactions, according to one example embodiment.

[0014] FIG. **7** is a flowchart illustrating a leaderless method for Messagenodes to reach consensus, according to one example embodiment.

[0015] FIG. **8** is a flowchart illustrating a method for a Validator to sign a block in a pre-commit phase, according to one example embodiment.

[0016] FIG. **9** is a flowchart illustrating a method for committing a block, according to one example embodiment.

[0017] FIG. **10** a block diagram illustrating components of an example machine able to read instructions from a machine-readable medium and execute them in a processor (or controller), according to one example embodiment.

DETAILED DESCRIPTION

[0018] The Figures (FIGS.) and the following description describe certain embodiments by way of illustration only. One skilled in the art will readily recognize from the following description that alternative embodiments of the structures and methods may be employed without departing from the principles described. Reference will now be made to several embodiments, examples of which are illustrated in the accompanying figures. It is noted that wherever practicable similar or like reference numbers are used in the figures to indicate similar or like functionality. Also, where similar elements are identified by a reference number followed by a letter, a reference to the number alone in the description that follows may refer to all such elements, any one such element, or any combination of such elements.

Overview

[0019] In various example embodiments, a consensus network uses a leaderless, asynchronous, probabilistic Byzantine consensus protocol to maintain a distributed ledger (e.g., a blockchain). The term blockchain is used herein for convenience but one of skill in the art will understand that other forms of distributed ledger may be used. Similarly, the term hash is used for convenience, but some embodiments may use other forms of reduced representation. The blockchain may be used to process and maintain a record of transactions (e.g., financial transactions) between users. By using a pipelined and parallelized approach for reaching consensus, the consensus network may achieve high scalability while maintaining high decentralization in the form of number of active processes participating in the consensus.

[0020] In one embodiment, a Messagenode receives transaction batches and broadcasts a message in the consensus